Git Basic Workflow

Git workflows refer to the structured way developers use Git to manage changes in a project, ensuring collaboration, consistency, and efficient version control. Below is a detailed breakdown of a **Git workflow**, from initialization to collaboration and deployment.

---

1. Initialize a Repository

A Git repository is the starting point. It can be created for a new project or initialized in an existing directory.

Example:

Step 1: Navigate to your project folder

cd my-project

Step 2: Initialize a Git repository

git init

This creates a .git folder in the project directory, which stores all Git-related metadata.

---

2. Create and Stage Files

Once the repository is initialized, files are created or modified and staged for tracking.

Example:

Step 1: Create a file

echo "Hello, World!" > hello.txt

Step 2: Stage the file

git add hello.txt

The git add command stages the file, preparing it to be committed. You can stage multiple files or an entire directory using:

git add .

---

3. Commit Changes

A commit saves the changes in the repository's history.

Example:

git commit -m "Initial commit: Add hello.txt"

- **-m**: Adds a message describing the changes.
- Commits represent a snapshot of the repository at a specific point.

---

4. Connect to a Remote Repository

To share your work, connect your local repository to a remote one (e.g., GitHub, GitLab).

Example:

Step 1: Add a remote repository

git remote add origin https://github.com/username/my-project.git

Step 2: Verify the remote repository

git remote -v

This associates your local repository with a remote repository named origin.

---

5. Push Changes to the Remote Repository

Upload commits from the local repository to the remote.

Example:

git push -u origin main

- **-u**: Sets the upstream branch so future pushes can be done with git push.
- This command uploads the main branch to the origin remote.